Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 5845094, 11 pages
https://doi.org/10.1155/2021/5845094

Research Article

Discriminative Codebook Hashing for Supervised Video Retrieval

Xiaoman Bian,[1] **Rushi Lan**,[1] **Xiaoqin Wang**,[1] **Chen Chen**,[1] **Zhenbing Liu**,[1] **Xiaonan Luo**,[1] **and Kuei-Kuei Lai**[2]

[1]*Guangxi Key Laboratory of Image and Graphic Intelligent Processing, Guilin University of Electronic Technology, Guilin 541004, China*
[2]*Department of Business Administration, Chaoyang University of Technology, Taichung 413310, Taiwan, China*

Correspondence should be addressed to Xiaoqin Wang; xqwang@guet.edu.cn and Kuei-Kuei Lai; laikk.tw@gmail.com

Received 18 May 2021; Accepted 12 August 2021; Published 25 August 2021

Academic Editor: Nian Zhang

In recent years, hashing learning has received increasing attention in supervised video retrieval. However, most existing supervised video hashing approaches design hash functions based on pairwise similarity or triple relationships and focus on local information, which results in low retrieval accuracy. In this work, we propose a novel supervised framework called discriminative codebook hashing (DCH) for large-scale video retrieval. The proposed DCH encourages samples within the same category to converge to the same code word and maximizes the mutual distances among different categories. Specifically, we first propose the discriminative codebook via a predefined distance among intercode words and Bernoulli distributions to handle each hash bit. Then, we use the composite Kullback–Leibler (KL) divergence to align the neighborhood structures between the high-dimensional space and the Hamming space. The proposed DCH is optimized via the gradient descent algorithm. Experimental results on three widely used video datasets verify that our proposed DCH performs better than several state-of-the-art methods.

1. Introduction

Under the condition of the increase in smartphones, the amount of video data has shown an explosive growth trend [1–3]. For example, TikTok has over 400 million daily active users who upload approximately 2,000 videos every minute. YouTube receives a total of 100 hours of videos per minute [4–6]. Due to the economic storage and efficiency of binary codes, hash-based methods have been widely applied to visual retrieval tasks [7–13].

Previous hash-related work [14] mainly focused on image hashing and can be divided into data-independent and data-dependent methods. Data-independent approaches learn binary codes without data information but through random space projection. The most representative algorithm is local sensitive hashing (LSH) [15], which generates huge redundant information using random mapping and obtains satisfactory performance with long hash codes. Data-dependent hash methods [16–18], which can also be divided into unsupervised hashing and supervised hashing, are proposed to generate more efficient hash codes by maintaining the neighborhood structure between data. For example, Gong et al. [19] proposed iterative quantization hashing (ITQ), which minimizes quantization error by rotating principal component analysis (PCA) projection data. Spectral hashing (SH) [20] assumes that data obey a uniform distribution and divides the data according to the main direction of the data stream. Density sensitive hashing (DSH) [21] extends LSH by studying structural information. Zhang et al. [22] developed a convergence-preserving parametric learning algorithm, called latent factor hashing (LFH), to learn similarity-preserving binary codes based on latent factor models. Liu et al. [23] proposed kernel supervised hashing (KSH) by applying kernel-based formulas to accommodate linearly inseparable data and designed a greedy algorithm to solve the hash function optimization problem.

In recent years, hashing methods proposed for video retrieval have also received extensive attention [24–31] and are composed of two categories: machine learning methods and deep hashing. Machine learning methods, resembling image hashing approaches, learn binary codes of video

keyframes based on the low-level manual features and then calculate video hashing codes via averaging. Wu et al. [4] employed video hashing via using color histograms to obtain global features. This is the first application of hash learning in the video field. Multiple-feature hashing (MFH) [32] adopts the weight-based method to combine different features. Ye et al. [33] used video structural information in the supervised learning paradigm to obtain the optimal binary codes. Stochastic multiview hashing (SMVH) [34] attempts to separately calculate the probability similarity matrices of video frames in the feature space and the Hamming space, and then, the difference between the above two probability matrices is minimized using the KL divergence. Nie et al. [35] defined joint multiview hashing (JMVH) by maximizing the interclass distance and minimizing the innerclass distance to preserve the global structure and local structure with multiple features. Boosting temporal video hashing (BTVH) [36] studies the multitable learning problem to boost the performance and captures the inherent similarity of video from both visual and temporal perspectives. In addition, some researchers in recent years have used deep networks to obtain the temporal and spatial information between keyframes. For instance, central similarity quantization (CSQ) [37] learns the temporal information by using 3D convolutional neural networks and proposes a view point called hash center to enhance the central similarity.

However, most existing video hashing approaches may lead to the following problems. (1) Low discriminability among different categories: functions based on pairwise similarity or triple relationships only consider local information, which results in good maintenance of the information of similar samples but shows poor performance in distinguishing samples from different categories. (2) Poor performance in real-world scenarios: in real application scenarios, similar data often accounts for only a small proportion, and most samples are not similar, which leads to low efficiency when the data are imbalanced [37]. (3) Greater time costs on deep learning: deep learning frameworks are time-consuming when training models and have no significant performance based on the spatiotemporal information extracted by the network. Hence, these video hashing functions cannot learn discriminative hash codes to enhance the performance.

To solve the above problems, in this work, we propose a novel framework for supervised video retrieval, called discriminative codebook hashing, which considers the global structure to construct the hash function. DCH encourages samples within the same category to converge to the identical codeword and maximizes the mutual distances between different categories. Specifically, the discriminative codebook is first generated based on two characters: the predefined distance between intercode words and Bernoulli distributions for ensuring that each hash bit stores more information. Then, to keep the similarity matrix between the feature space and the Hamming space, the composite KL divergence is proposed to solve this problem. Finally, the gradient descent algorithm is utilized to optimize the algorithm. In this way, we can obtain discriminative binary codes for video retrieval. Figure 1 shows the framework of DCH, and the method we proposed has the following innovations:

(i) We proposed the discriminative codebook based on the predefined distance between intercode words and Bernoulli distributions for ensuring each hash bit to store more information

(ii) The DCH method, which can maximize the distance of the intercode words generated by the predefined codebook to learn discriminative binary codes for supervised video retrieval, is proposed

(iii) We verify our proposed method by experimenting on three widely used datasets, which shows that DCH has a significant improvement in contrast with several state-of-the-art methods

The other sections are organized as follows. Section 2 introduces some preliminary works. Section 3 introduces the proposed discriminative codebook hashing in detail. The experimental work is presented in Section 4, and the conclusion of DCH is shown in Section 5.

2. Preliminary Work

In this section, we briefly introduce the preliminary work, namely, stochastic multiview hashing [34]. It is a supervised video retrieval method that aims to preserve the similarity structure from the original space to the Hamming space.

Let $V = \{v_i\}_{i=1}^{n_v}$ be the video set, where v_i indicates the ith video of V and n_v is the number of videos. $H = \{h_i\}_{i=1}^{n_v}$ is hash code of the video set, where $h_i \in \{0, 1\}$ is l-bit length binary codes transformed by v_i. The video features are extracted based on the set of keyframe features $X = \{x_i\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^{1\times d}$, n is the number of keyframes, and d is the dimension of each keyframe. $Z = \{z_i\}_{i=1}^{n}$ represents the corresponding binary codes of the keyframes, where $z_i \in \mathbb{R}^{1\times l}$. The conversion relationships between the above variables are formulated as

$$\widetilde{Z} = XW + b, \tag{1}$$

$$Z = \text{sigmoid}(\widetilde{Z}), \tag{2}$$

$$h_i = T\left(\frac{1}{|\text{Ind}_i|}\sum_{j\in \text{Ind}_i} z_{j\cdot}\right), \tag{3}$$

where $\widetilde{Z} \in \mathbb{R}^{n\times l}$ is the temporal result of linear projection, $b \in \mathbb{R}^l$ is a bias parameter, $W \in \mathbb{R}^{d\times l}$ is the projection matrix, Ind_i is the set of frames, and $|\text{Ind}_i|$ is the sum of samples in the set. The high-dimensional keyframe feature matrix X is first projected into the lower matrix $\widetilde{Z}$. Then, the sigmoid function is used to map the variable between 0 and 1. Finally, a thresholding function is used to change the data into a binary code with $T(y) = 0$ if $y < 0.5$ and $T(y) = 1$, otherwise.

SMVH keeps the similarity matrix between the feature space and the Hamming space using a composite KL divergence measure. In particular, it separately calculated the similarity probability matrix P in the original space and the

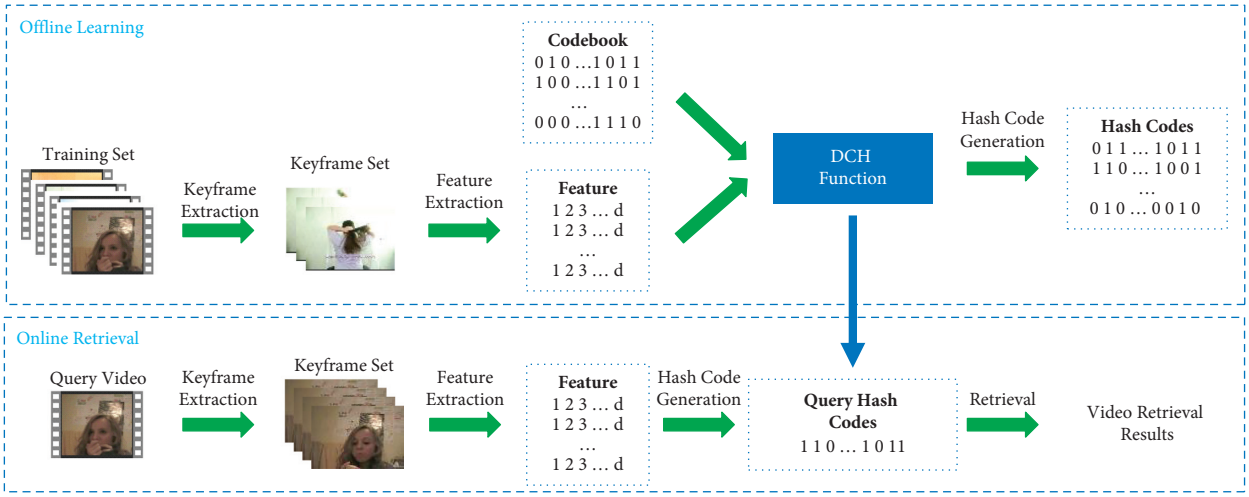

FIGURE 1: The framework of DCH. We divide the entire experiment into two steps, namely, offline learning and online retrieval. In the offline phase, we join keyframe features and predefined codebook to learn hash functions. In the online phrase, we map the query video into a set of binary codes through hash functions. Next, we use the exclusive or (XOR) operation to obtain the Hamming distance between the query video and samples in the database. Finally, we take videos with the shortest Hamming distance as the video retrieval results.

pairwise similarity matrix Q among samples in the Hamming space. Then, the KL divergence is used to examine how well the above two probability matrices P and Q match. Therefore, the objective function of SMVH is defined as follows:

$$\min_{W,b} S_{\mathrm{KL}}(W,b) + \frac{\mu}{2}\|W\|_F^2, \tag{4}$$

where $\mu > 0$ controls the weight of the regular term to prevent overfitting and $S_{\mathrm{KL}}(W,b)$ is the composite KL divergence. The latter can be represented as

$$S_{\mathrm{KL}}(W,b) = \lambda \mathrm{KL}(P \| Q) + (1-\lambda)\mathrm{KL}(Q \| P), \tag{5}$$

where $0 \le \lambda \le 1$ controls the influence of the composite KL divergence, $P = \{p_i\}_{i=1}^n \in \mathbb{R}^{n\times n}$ is the similarity structure based on X, and $Q = \{q_i\}_{i=1}^n \in \mathbb{R}^{n\times n}$ is another probability matrix preserving the similarity information of Z in the Hamming space. In addition, the KL divergence is defined as follows:

$$\mathrm{KL}(P \| Q) = \sum_{i=1}^{n} \sum_{j \ne i} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}, \tag{6}$$

where $p_{j|i}$ is a conditional probability that reflects the similarity between x_i and x_j, and another conditional probability $q_{j|i}$ represents the probability of returning z_j given the query z_i.

3. Discriminative Codebook Hashing

In this section, we present the proposed DCH in detail through four parts, including the proposed discriminative codebook, the objective function, algorithmic optimization, and complexity analysis.

3.1. Discriminative Codebook. Motivated by CSQ [37], we propose a novel and discriminative codebook $C = \{c_i\}_{i=1}^m$ for supervised video retrieval, where $c_i \in \{0,1\}^{1\times l}$ is the code word of the ith category. The proposed codebook is defined according to two characters. The first is that the value in the same bit of different code words obeys a Bernoulli distribution. Specifically, the proportions of 0 and 1 of the same bit in different categories are both 50%, that is, $c_{\cdot i}$ has a 50% probability of being 0 or 1, which will maximize the entropy and store more information in each bit. The other is that the mutual distances among intercode words are defined as follows:

$$D_H(c_i, c_j) \ge \frac{l}{2} - f, \tag{7}$$

where D_H is the Hamming distance between code words c_i and c_j, l is the length of binary codes, and f represents the fault tolerance. The mutual distance between intercode words will be the largest constrained by equation (7).

Overall, the proposed codebook encourages samples within the same category to converge to the same codeword and maximizes the mutual distance between different categories. Therefore, the proposed codebook can preserve global structures and help generate discriminative binary codes for video retrieval. The scheme of the proposed discriminative codebook is presented in Algorithm 1.

3.2. Objective Function. According to the proposed discriminative codebook C, we expand each row of the codebook matrix C into $R = \{r_i\}_{i=1}^n$ according to the number of samples, where $r_i \in \mathbb{R}^{1\times l}$. The detailed generation process of R is shown in Algorithm 2. We minimize the error between the binary codes and the predefined codebook as

Input: the number of categories m; the number of samples per category n_i; code length l; maximum number of iterations T_c; fault tolerance rate f.
Output: codebook $C \in R^{m \times l}$
(1) **for** iteration $t_c = 1 : T_c$
(2) **for** category $i = 1 : m$
(3) $c_{.i}$[random half coordinate] = 1
(4) $c_{.i}$[the rest coordinate] = 0
(5) **end**
(6) **if** any two rows of C satisfy equation (7)
(7) break
(8) **end**
(9) **end**

Algorithm 1: Discriminative codebook.

Input: training data $X \in \mathbb{R}^{n \times d}$; codebook $C \in \mathbb{R}^{m \times l}$; maximum number of iterations T; code length l; parameters λ, μ, γ; learning rate α;
Output: hash codes $H \in \{0, 1\}^{n_v \times l}$.
(1) **Initialization**: initialize the projection matrix W and bias matrix b as a random matrix and vector.
(2) **Generating R according to the number of samples**:
(3) **for** category $i = 1 : m$
(4) $R = [R; \text{repmat}(C(i, :), n_i, 1)]$
(5) **end**
(6) **Gradient descent**:
(7) **for** iteration $i = 1 : T$
(8) **W-Step**: $W^{(i+1)} = W^{(i)} + \alpha dW$
(9) **b-Step**: $b^{(i+1)} = b^{(i)} + \alpha db$
(10) **end**
(11) **Video binary code computation**: video hash codes are obtained by equations (1)–(3).

Algorithm 2: Discriminative codebook hashing.

$$\min_{W,b} \|Z - R\|_F^2. \tag{8}$$

Specifically, for each $z_i \in Z$, we take r_i as the codebook of $z_i \in Z$ to make samples in the same category share the same codebook and samples in different categories have discriminative binary codes.

To keep the similarity matrix between the feature space and the Hamming space, we join the composite KL divergence and our proposed codebook to construct the overall objective function of DCH as follows:

$$\min_{W,b} S_{\text{KL}}(W, b) + \frac{\gamma}{2}\|Z - R\|_F^2 + \frac{\mu}{2}\|W\|_F^2, \tag{9}$$

where γ controls the weight of the error loss between the codebook and the learned hash codes, and the second term of equation (9) aligns values between binary codes and their corresponding code word.

In this way, our proposed DCH can solve the problem that other algorithms only consider the pairwise relationships and ensure that samples in the same category share the same code word. Furthermore, DCH maximizes the mutual distances between different categories and then obtains discriminative binary codes.

3.3. Algorithmic Optimization. The optimization problem has two main variables: W and b. Our solution is to use the gradient descent algorithm to find good solutions. To facilitate the writing, we split the objective function equation (9) into three parts:

$$\begin{aligned} \Phi_1(W, b) &= S_{KL}(W, b), \\ \Phi_2(W, b) &= \frac{\gamma}{2}\|Z - R\|_F^2, \\ \Phi_3(W) &= \frac{\mu}{2}\|W\|_F^2. \end{aligned} \tag{10}$$

The detailed optimization procedure is presented as follows.

W-Step: the corresponding problem is to minimize the following loss function:

$$\min_{W} S_{\mathrm{KL}}(W,b) + \frac{\gamma}{2}\|Z-R\|_F^2 + \frac{\mu}{2}\|W\|_F^2. \tag{11}$$

To compute the optimal W, the relevant deviation formula can be expressed as

$$\mathrm{d}W = \frac{\partial \Phi_1(W,b)}{\partial W} + \frac{\partial \Phi_2(W,b)}{\partial W} + \frac{\partial \Phi_3(W)}{\partial W}. \tag{12}$$

The derivative of $\partial\Phi_1(W,b)$ w.r.t. W can be computed as follows:

$$\frac{\partial \Phi_1(W,b)}{\partial W} = \left[\frac{\partial \Phi_1(W,b)}{\partial z_{ik}}\frac{\partial z_{ik}}{\partial w_{kj}}\right]_{d\times l}, \tag{13}$$

where $\partial\Phi_1(W,b)/\partial z_{ik}$ and $\partial z_{ik}/\partial w_{kj}$ are represented as

$$\frac{\partial \Phi_1(W,b)}{\partial z_{ik}} = 2\left(\lambda\left(p_{i|t} - q_{i|t} + p_{t|i} - q_{t|i}\right) + (1-\lambda) * \left(q_{t|i}\sum_{g\neq i} q_{g|i}\log\frac{q_{g|i}}{p_{g|i}} + q_{i|t}\sum_{g\neq t} q_{g|t}\log\frac{q_{g|t}}{p_{g|t}} - \log\frac{q_{t|i}}{p_{t|i}} - \log\frac{q_{i|t}}{p_{i|t}}\right)(z_{ik}-z_{tk}),\right.$$

$$\left.\frac{\partial z_{ik}}{\partial w_{kj}} = z_{ik}(1-z_{ik})x_{ji}.\right. \tag{14}$$

Following the norm derivation law, $\partial\Phi_2(W,b)/\partial W$ can be optimized as follows:

$$\frac{\partial \Phi_2(W,b)}{\partial W} = \frac{\partial \Phi_2(W,b)}{\partial Z}\frac{\partial Z}{\partial W} = X^T((Z-R)\odot(Z\odot(1-Z))), \tag{15}$$

where $\odot$ indicates that the elements in the same position of two matrices are multiplied.

For $\partial\Phi_3(W)/\partial W$, we have the derivative that

$$\frac{\partial \Phi_3(W)}{\partial W} = \mu W. \tag{16}$$

b-Step: the subproblem of b is given by

$$\min_{b} S_{KL}(W,b) + \frac{\gamma}{2}\|Z-R\|_F^2. \tag{17}$$

The deviation w.r.t. b can be expressed as

$$\mathrm{d}b = \frac{\partial \Phi_1(W,b)}{\partial b} + \frac{\partial \Phi_2(W,b)}{\partial b}. \tag{18}$$

The derivative of $\partial\Phi_1(W,b)/\partial b$ is described as follows:

$$\frac{\partial \Phi_1(W,b)}{\partial b} = \left[\frac{\partial \Phi_1(W,b)}{\partial z_{ik}}\frac{\partial z_{ik}}{\partial b_k}\right]_{1\times l}, \tag{19}$$

where

$$\frac{\partial z_{ik}}{\partial b_k} = z_{ik}(1-z_{ik}). \tag{20}$$

The second term of equation (18) is described as follows:

$$\frac{\partial \Phi_2(W,b)}{\partial b} = \frac{\partial \Phi_2(W,b)}{\partial Z}\frac{\partial Z}{\partial b} = (Z-R)\odot(Z\odot(1-Z)). \tag{21}$$

Algorithm 2 describes the overall algorithm optimization process of the proposed DCH.

3.4. Complexity Analysis. The time complexity of the entire training process of SMVH [34] is approximately $O(Tn^3 + n^2)$, and the proposed DCH algorithm adds two parts time-consuming on this basis. The first part is the learning process of C, and the time complexity is $O(T_c l)$. The second part is that the time complexity of optimizing equations (15) and (21) together is $O(\mathrm{d}nl)$ in each iteration. Therefore, the overall time complexity of DCH is $O(n^2 + T_c l + T(n^3 + \mathrm{d}nl))$. In this work, time complexities $O(T_c l)$ and $O(\mathrm{d}nl)$ can be ignored due to T_c, $l, d \ll n$ so that our complexity is nearly $O(Tn^3 + n^2)$. Additionally, the calculation of the hash codes is a linear projection with a time complexity of approximately $O(1)$, and the online search can be performed by XOR operations. Although the algorithm proposed in this paper adds a constraint on SMVH, the maximum number of iterations T directly affects the time complexity of the algorithm. It can be proven in subsequent experiments that DCH can converge in fewer iterations. Thus, the time complexity of DCH is in a reasonable range.

4. Experiments

In this section, we first introduce the datasets used in this paper, and then, the baselines and some experimental details will be introduced. Finally, we present the experimental results.

4.1. Datasets. CC_WEB_VIDEO [4] is the most useful dataset in near-duplicate video retrieval (NDVR) research, which contains data from YouTube, Google, and Yahoo. There are 12,877 videos that are divided into 24 sets, and keyframes are extracted by a uniform sampling method to represent the video. Since some videos do not have label information, we take 3,482 videos with labels as the experimental dataset. In each category, we select 70% of the video data as the training set and the remainder as the testing set. We extract 10 keyframes for each video uniformly and

extract 4096-dimensional features to represent keyframes by using the pretrained VGG-19 network.

HMDB51 [38] contains 6,766 human action videos selected from movies and some other public sources such as YouTube. The dataset is divided into 51 categories, and each of them includes approximately 100 clips. In each category, we randomly select 45 video samples. Of these, 25 videos are added to the training set and the rest are select to the testing set. We uniformly extract 10 keyframes for each video, and the VGG-19 pretraining network is used to extract the 4096-dimensional deep features.

UCF101 [39] contains 13,320 videos which has been divided into 101 human behavior categories, such as sports, instruments, character interactions, and others used for action recognition. We randomly select 70 videos in each category to join the training set, and 30 videos to join the testing set. For each video, 10 keyframes are uniformly selected to represent the video. We use VGG-19 to extract the 4096-dimensional features for each keyframe.

4.2. Experimental Setting

4.2.1. Baselines. Several state-of-the-art hash functions, including ITQ [19], SH [20], DSH [21], LFH [22], KSH [23], JMVH [35], and SMVH [34], are used for comparison. Among these methods, ITQ, SH, and DSH are unsupervised hashing methods, while LFH, KSH, JMVH, and SMVH are supervised hashing methods. For the comparative test, we use the source codes published to conduct the experiment. JMVH and SMVH can also be used for multiview video retrieval, but in this paper, we only test these methods as a single view method. It is worth noting that all the experimental results are obtained in MATLAB R2016a on the same computer with an Intel Core i7-6700 CPU @ 3.40 GHz, 72 GB RAM and the 64 bit Windows 10 operating system.

4.2.2. Evaluation Metrics. We use four popular evaluation metrics to comprehensively evaluate experimental results. The mean average precision (mAP) is widely used in the retrieval field. The higher the mAP score is, the better the retrieval performance of the method is. The precision@K curve represents the precision accuracy versus the first K retrieved samples, where precision represents the proportion of the number of retrieved correct videos to the total number of retrieved videos. The recall@K curve represents the average recall rate versus the first K retrieved samples, where recall represents the proportion of the correct video volume retrieved in all near-duplicate video samples. The precision-recall (PR) curve is an index used to evaluate reliability and is widely used in the fields of medicine and machine learning.

4.2.3. Parameter Selection. We have three model parameters, including λ, μ, and γ, and the number of iterations T. According to SMVH [34], we set $\lambda = 0.9$ and $\mu = 0.01$. As shown in Figure 2(a), when γ is in the range of 0.05 to 1, the results are stable across three different datasets. Therefore, we empirically choose $\gamma = 1$ in our proposed model. The maximum number of iterations T determines the training time cost and the performance, so it is worth discussing. Figure 2(b) shows the effect of the maximum iterations T in the range of 100 to 1400 on mAP performance. For HMDB51, it can be seen that the best mAP is generated with $T = 800$ before decreasing. However, in the other two datasets, $T = 800$ is not an optimal experimental result. Therefore, after comprehensive consideration, $T = 1000$ is set as the final parameter setting.

4.3. Results and Discussion. Table 1 shows the mAP results for different lengths of hash codes on the three datasets, and the results of other evaluation metrics are shown in Figures 3–5. We will give the detailed analysis of all results of the three datasets in the following parts.

According to Table 1, for the CC_WEB_VIDEO dataset, the mAPs are very high because the dataset is movie clips, and videos of the same category are near-duplicate videos. As shown in Table 1, the performance of the proposed DCH is at least 1.85% better than that of the other methods from 32 to 64 bits. When the code length is 96 bits, the mAP of DCH is slightly lower than that of LFH. As shown in Figure 3, the experimental results of our method in precision@K and recall@K are equal to or slightly higher than those of most other methods. Besides, as the code length increases, the performance of our proposed DCH gradually surpasses that of other methods. Figures 3(i)–3(l) show that the area surrounded by DCH is gradually increasing.

Table 1 shows that our proposed DCH performs better than other hash methods in most cases in the HMDB51 dataset. Although the mAP performance of the JMVH method surpasses 2.39% over that of DCH with 32 bits, the mAPs of our proposed DCH are better than those of the other comparison methods in the subsequent experiments. Figure 4 shows that when the length of hash codes is larger than 32 bits, regardless of whether precision@K curve, recall@K curve, or PR curve is used, DCH has excellent performance compared with other methods in all metrics for the precision@K curve, recall@K curve, and PR curve.

For the UCF101 dataset, DCH obtained the optimal experimental results in the range of [32, 48, 64] bits. It is worth noting that the size of the UCF101 dataset is relatively large, and SMVH cannot obtain discriminative video hash when the hash code length is very small. Therefore, SMVH has no experimental results available for $l = 32$ and $l = 48$. As shown in Figure 5, the performance of DCH is much higher than those of some of the methods except JMVH. We can see that the recall rate of DCH for positive samples is slightly lower than that of JMVH based on Figures 5(e)–5(h). Figures 5(i)–5(k) show that the performance of DCH for 32 to 48 bits is better than those of all other methods for the PR curve.

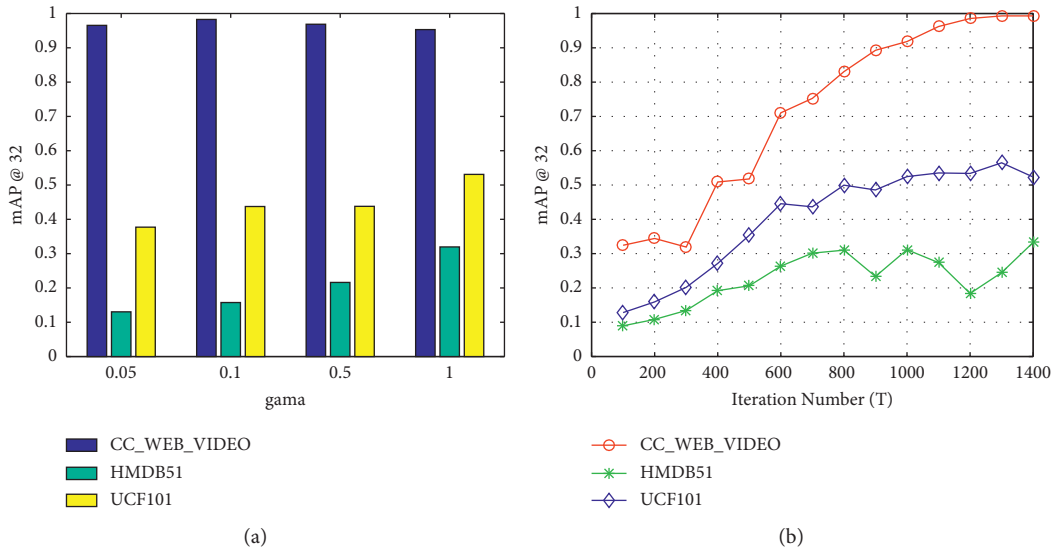

Figure 2: Parameter analysis on the CC_WEB_VIDEO, HMDB51, and UCF101 datasets. (a) mAP vs. γ (weight parameter γ) and (b) mAP vs. T (iteration parameter T).

Table 1: The mAP of different hash code lengths on three datasets, where the best experimental results are given in bold.

Method	CC_WEB_VIDEO				HMDB51				UCF101			
	32 bits	48 bits	64 bits	96 bits	32 bits	48 bits	64 bits	96 bits	32 bits	48 bits	64 bits	96 bits
ITQ [19]	0.6877	0.7725	0.8099	0.7700	0.0697	0.0749	0.0793	0.0885	0.1383	0.1620	0.1801	0.2119
SH [20]	0.6729	0.7026	0.6994	0.6708	0.0662	0.0657	0.0642	0.0653	0.1033	0.1138	0.1244	0.1395
DSH [21]	0.6510	0.7060	0.6929	0.8158	0.0505	0.0628	0.0671	0.0750	0.0720	0.0667	0.0815	0.1082
LFH [22]	0.8327	0.8088	0.9854	**0.9912**	0.0141	0.0208	0.0148	0.0225	0.0032	0.0038	0.0078	0.0113
KSH [23]	0.9368	0.9030	0.9477	0.8761	0.2470	0.2811	0.3054	0.3144	0.3222	0.3598	0.3972	0.4075
JMVH [35]	0.7842	0.5576	0.4335	0.3745	**0.2807**	0.3015	0.2418	0.1295	0.3941	0.5166	0.6007	**0.6875**
SMVH [34]	0.9346	0.9411	0.9543	0.7490	0.1212	0.1399	0.1374	0.0319	—	—	0.0094	0.0304
DCH	**0.9531**	**0.9763**	**0.9886**	0.9858	0.2568	**0.3819**	**0.3600**	**0.4150**	**0.5310**	**0.6137**	**0.6609**	0.6458

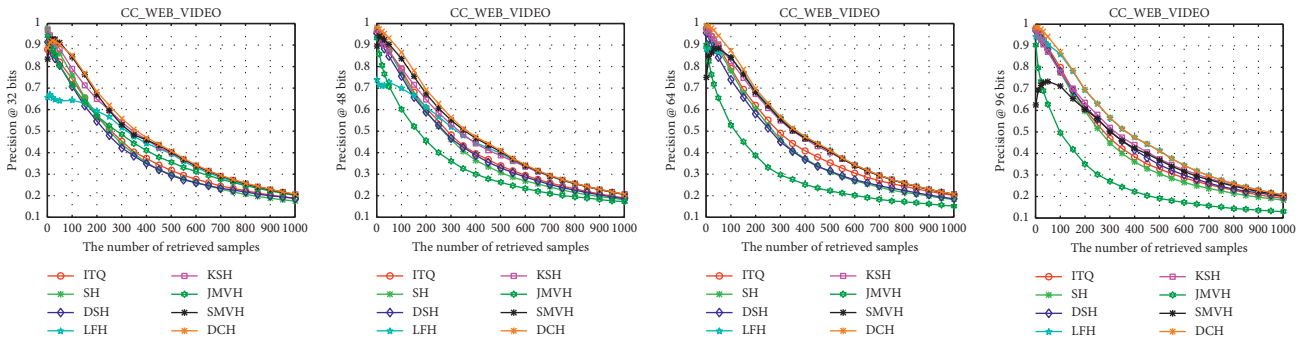

Figure 3: Continued.

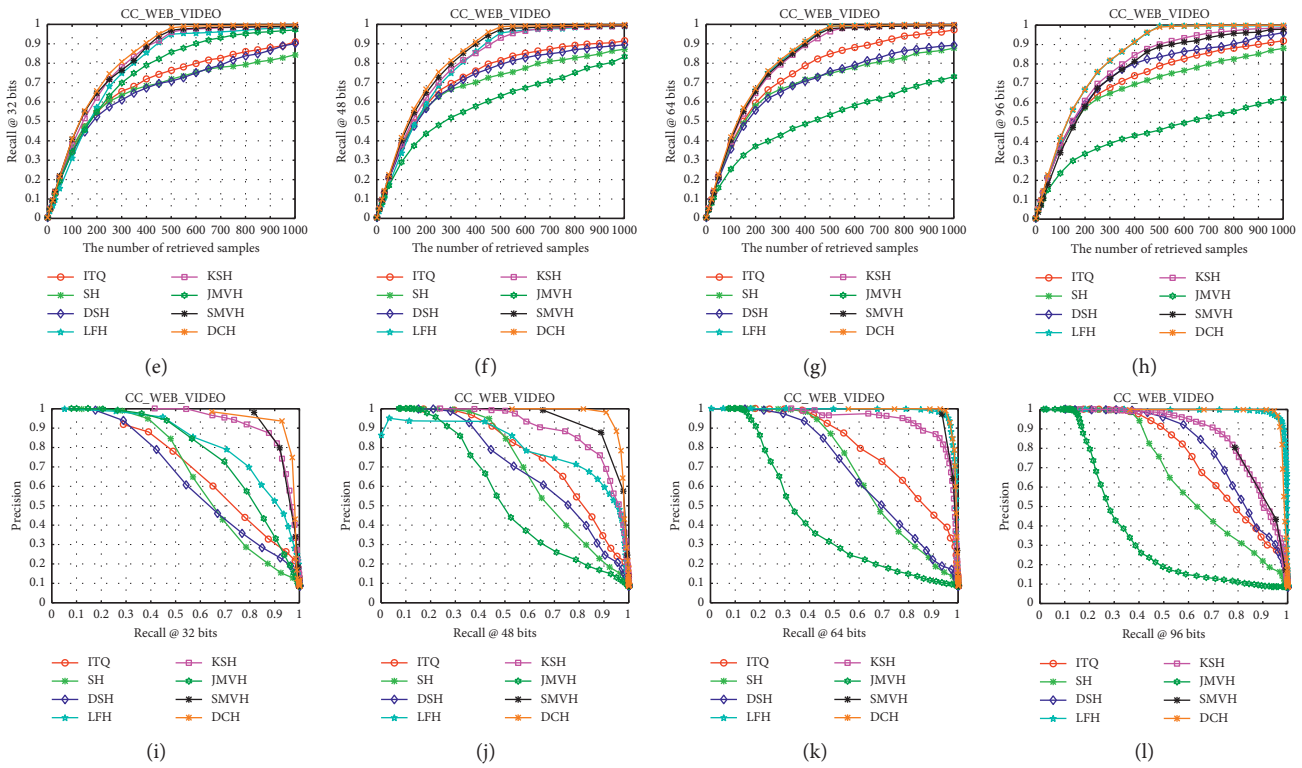

Figure 3: Precision@K (a–d), recall@K (e–h), and PR (i–l) curves on the CC_WEB_VIDEO dataset.

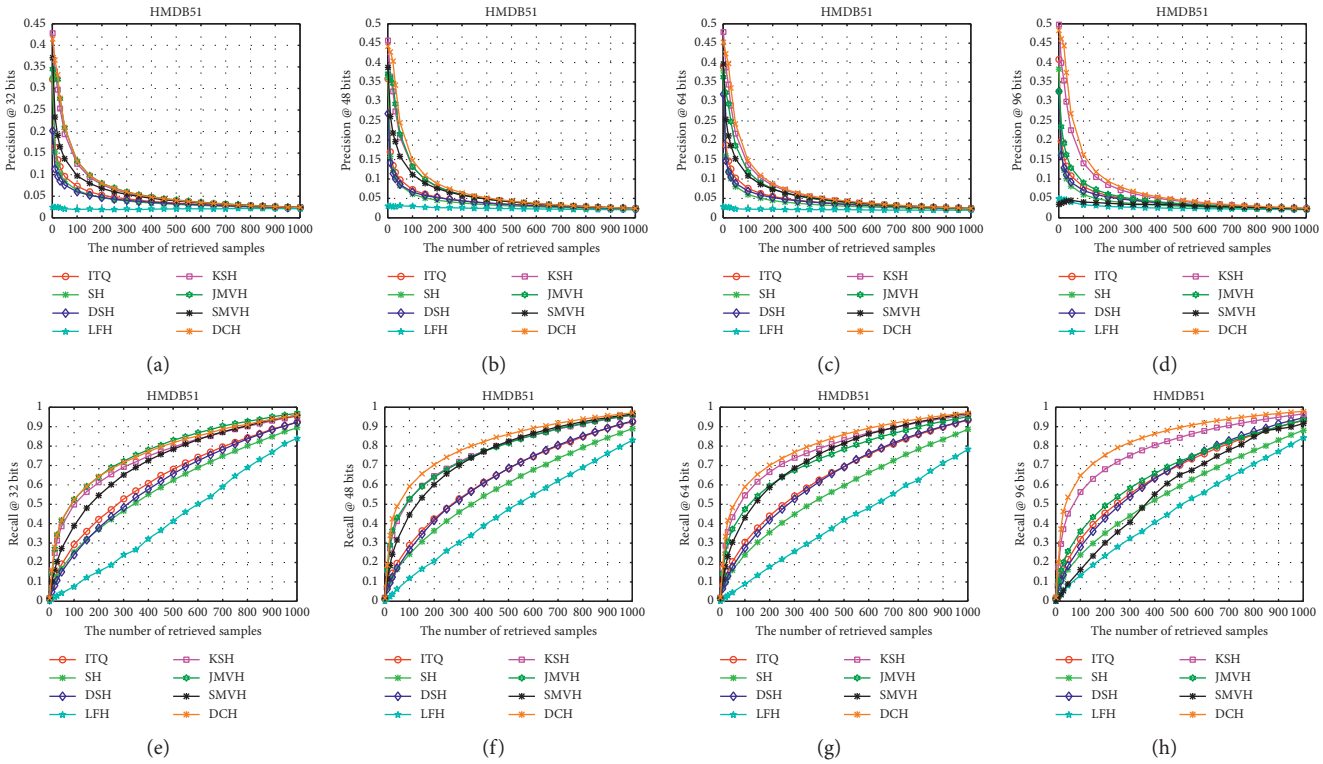

Figure 4: Continued.

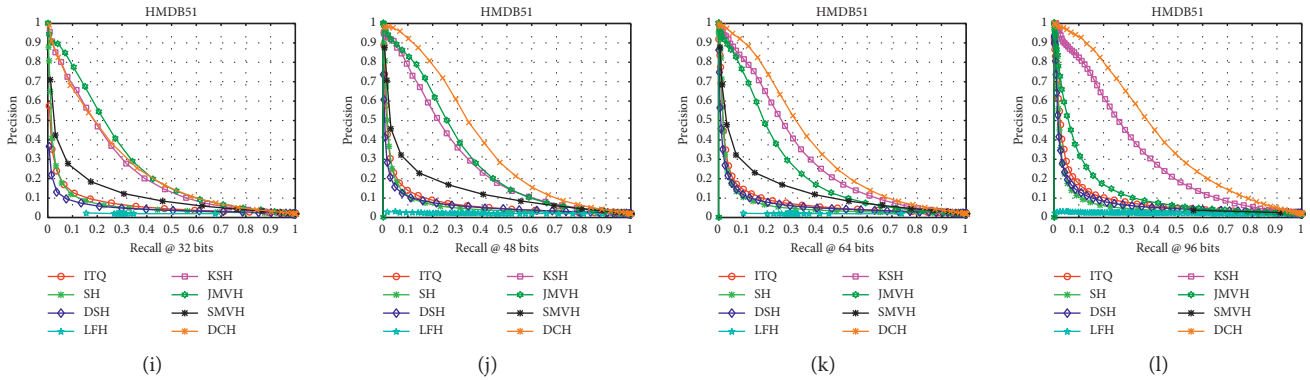

Figure 4: Precision@K (a–d), recall@K (e–h), and PR (i)–(l) curves on the HMDB51 dataset.

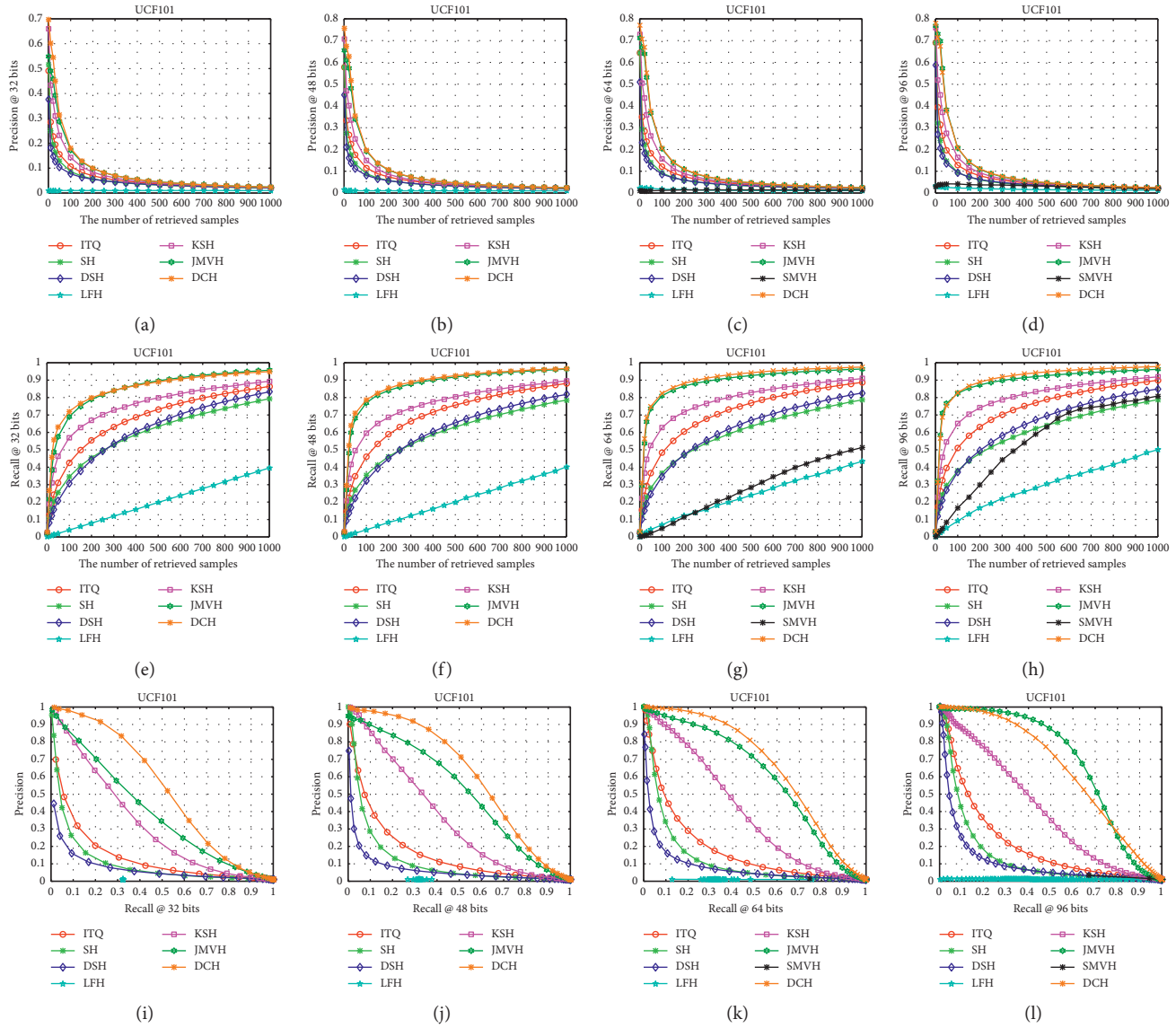

Figure 5: Precision@K (a–d), recall@K (e–h), and PR (i–l) curves on the UCF101 dataset.

5. Conclusion

In this paper, we propose a novel supervised video hashing framework, termed discriminative codebook hashing, which can generate discriminative binary codes for video retrieval. The proposed DCH encourages samples within the same category to converge to the same code word and maximizes the mutual distances between different categories. Specifically, we generate a discriminative codebook to distinguish between samples of different categories more accurately. Extensive experimental results prove that the performance of DCH is significantly improved compared to several state-of-the-art methods. In future work, we will use a smaller matrix storing the similarity information between samples to avoid consuming considerable training time and space when the amount of data is large. This will improve the performance of the model while reducing the time complexity.

Data Availability

CC_WEB_VIDEO dataset can be downloaded from http://vireo.cs.cityu.edu.hk/webvideo/, the HMDB51 dataset can be downloaded from https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/#dataset, and the UCF101 dataset can be downloaded from https://www.crcv.ucf.edu/data/UCF101.php.

Conflicts of Interest

The authors declare that there are no conflicts of interest in the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (nos. 61902087, 61772149, 61936002, and 6202780103), Guangxi Science and Technology Project (nos. 2019GXNSFFA245014, AD18281079, AA18118039, and AD18216004), and Guangxi Key Laboratory of Image and Graphic Intelligent Processing (no. GIIP2001).

References

[1] X. Wu, C.-W. Ngo, A. G. Hauptmann, and H.-K. Tan, "Real-time near-duplicate elimination for web video search with content and context," *IEEE Transactions on Multimedia*, vol. 11, no. 2, pp. 196–207, 2009.

[2] V. O. Maraghi and K. Faez, "Scaling human-object interaction recognition in the video through zero-shot learning," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 9922697, 15 pages, 2021.

[3] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.

[4] X. Wu, A. G. Hauptmann, and C. Ngo, "Practical elimination of near-duplicates from web video search," in *Proceedings of the 15th International Conference on Multimedia*, pp. 218–227, ACM, Bavaria, Germany, 2007.

[5] Z. Lu, Y. Wang, Y. Li, X. Tong, C. Mu, and C. Yu, "Data-driven many-objective crowd worker selection for mobile crowdsourcing in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 30, p. 1, 2021.

[6] Y.-N. Ma, Y.-J. Gong, C.-F. Xiao, Y. Gao, and J. Zhang, "Path planning for autonomous underwater vehicles: an ant colony algorithm incorporating alarm pheromone," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 141–154, 2019.

[7] G.-H. Liu and Z. Wei, "Image retrieval using the fused perceptual color histogram," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8876480, 10 pages, 2020.

[8] R. Lan, L. Sun, Z. Liu, H. Lu, C. Pang, and X. Luo, "MaDNet: a fast and lightweight network for single-image super resolution," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1443–1453, 2021.

[9] W. Li, Y. Zhang, Y. Sun et al., "Approximate nearest neighbor search on high dimensional data-experiments, analyses, and improvement," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1475–1488, 2020.

[10] R. Lan, Y. Zhou, Z. Liu, and X. Luo, "Prior knowledge-based probabilistic collaborative representation for visual recognition," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1498–1508, 2020.

[11] X. Wang, R. Lan, H. Wang, Z. Liu, and X. Luo, "Fine-grained correlation analysis for medical image retrieval," *Computers & Electrical Engineering*, vol. 90, Article ID 106992, 2021.

[12] L. Shang, L. Yang, F. Wang, K. Chan, and X. Hua, "Real-time large scale near-duplicate web video retrieval," in *Proceedings of the 18th International Conference on Multimedia*, pp. 531–540, ACM, Firenze, Italy, 2010.

[13] N. Q. Ly, T. K. Do, and B. X. Nguyen, "Large-scale coarse-to-fine object retrieval ontology and deep local multitask learning," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 1483294, 40 pages, 2019.

[14] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *Proceedings of the 19th International Conference on Multimedia*, pp. 423–432, ACM, Scottsdale, AZ, USA, 2011.

[15] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the 20th ACM Symposium on Computational Geometry*, pp. 253–262, ACM, Brooklyn, NY, USA, 2004.

[16] W. Liu, J. Wang, S. Kumar, and S. Chang, "Hashing with graphs," in *Proceedings of the 28th International Conference on Machine Learning*, pp. 1–8, Bellevue, WA, USA, 2011.

[17] Y. Fang and Y. Ren, "Supervised discrete cross-modal hashing based on kernel discriminant analysis," *Pattern Recognition*, vol. 98, Article ID 107062, 2020.

[18] X. Liu, X. Nie, X. Xi, L. Zhu, and Y. Yin, "MoBoost: a self-improvement framework for linear-based hashing," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 871–880, ACM, Beijing, China, 2019.

[19] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.

[20] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proceedings of the 21st International Conference on Neural Information Processing Systems*, pp. 1753–1760, Vancouver, BC, USA, 2008.

[21] Z. Jin, C. Li, Y. Lin, and D. Cai, "Density sensitive hashing," *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1362–1371, 2014.

[22] P. Zhang, W. Zhang, W. Li, and M. Guo, "Supervised hashing with latent factor models," in *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 173–182, ACM, New York, NY, USA, 2014.

[23] W. Liu, J. Wang, R. Ji, Y. Jiang, and S. Chang, "Supervised hashing with kernels," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2074–2081, IEEE Computer Society, Providence, RI, USA, 2012.

[24] G. Wu, J. Han, Y. Guo et al., "Unsupervised deep video hashing via balanced code for large-scale video retrieval," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1993–2007, 2019.

[25] Y. Hao, T. Mu, J. Y. Goulermas, J. Jiang, R. Hong, and M. Wang, "Unsupervised t-distributed video hashing and its deep hashing extension," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5531–5544, 2017.

[26] G. Wu, L. Liu, Y. Guo et al., "Unsupervised deep video hashing with balanced rotation," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3076–3082, Sydney, Australia, 2017.

[27] V. E. Liong, J. Lu, Y.-P. Tan, and J. Zhou, "Deep video hashing," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1209–1219, 2017.

[28] S. Li, Z. Chen, J. Lu, X. Li, and J. Zhou, "Neighborhood preserving hashing for scalable video retrieval," in *Proceedings of the 2019 IEEE International Conference on Computer Vision*, pp. 8211–8220, IEEE, Seoul, South Korea, 2019.

[29] S. Chen, Y. Zhao, Q. Jin, and Q. Wu, "Fine-grained video-text retrieval with hierarchical graph reasoning," in *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10635–10644, IEEE, Seattle, WA, USA, 2020.

[30] R. Yang, F. Mentzer, L. V. Gool, and R. Timofte, "Learning for video compression with hierarchical quality and recurrent enhancement," in *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6627–6636, IEEE, Seattle, WA, USA, 2020.

[31] Y. Wang, X. Nie, Y. Shi, X. Zhou, and Y. Yin, "Attention-based video hashing for large-scale video retrieval," *IEEE Transactions on Cognitive and Developmental Systems*, 2019.

[32] J. Song, Y. Yang, Z. Huang, H. T. Shen, and J. Luo, "Effective multiple feature hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1997–2008, 2013.

[33] G. Ye, D. Liu, J. Wang, and S. Chang, "Large-scale video hashing via structure learning," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, pp. 2272–2279, IEEE Computer Society, Sydney, Australia, 2013.

[34] Y. Hao, T. Mu, R. Hong, M. Wang, N. An, and J. Y. Goulermas, "Stochastic multiview hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 1–14, 2017.

[35] X. Nie, W. Jing, C. Cui, C. J. Zhang, L. Zhu, and Y. Yin, "Joint multi-view hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 10, pp. 1951–1965, 2020.

[36] Y. Wu, X. Liu, H. Qin et al., "Boosting temporal binary coding for large-scale video search," *IEEE Transactions on Multimedia*, vol. 23, pp. 353–364, 2021.

[37] L. Yuan, T. Wang, X. Zhang et al., "Central similarity quantization for efficient image and video retrieval," in *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3080–3089, IEEE, Seattle, WA, USA, 2020.

[38] H. Kuehne, H. Jhuang, E. Garrote, T. A. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the 2011 IEEE International Conference on Computer Vision*, pp. 2556–2563, IEEE Computer Society, Barcelona, Spain, 2011.

[39] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: a dataset of 101 human actions classes from videos in the wild," *Computing Research Repository*, 2012, https://arxiv.org/abs/1212.0402.